

A new method to reduce the number of time delays in a network

Alexandre Wagemakers and Miguel A. F. Sanjuán

Departamento de Física, Universidad Rey Juan Carlos, Tulipán s/n, 28933 Móstoles, Madrid, Spain

(Dated: February 10, 2017)

Time delays may cause dramatic changes to the dynamics of interacting oscillators. Coupled networks of interacting dynamical systems can behave unexpectedly when the signal between the vertices are time delayed. It has been shown for a very general class of systems that the time delays can be rearranged as long as the total time delay over the constitutive loops of the network is conserved. This fact allows to reduce the number of time delays of the problem without loss of information. There is a theoretical lower bound for this number, but in many cases we can find a numerical solution that beats this limit. Here we propose a formulation of the problem and a numerical method to even further reduce the number of time delays in a network.

I. INTRODUCTION

Transmission delays are intrinsic to any process that exchange information. While in many applications these time delays are small enough to be neglected, in other cases they have a critical influence on the dynamics. Examples of connected dynamical systems appear frequently in physics, engineering and natural sciences [1–3]. A problem of interest in the dynamical systems community is the synchronization of coupled oscillators. Some progress has been made to understand the synchronization of oscillators when an identical time delay is present on every connection of a coupled system [4–6]. Otherwise, the problem of synchronization with nonidentical time delays spreaded accross a network is still open. In an effort to simplify the analysis of such networks, a new method called componentwise time-shift transformation [7] has been developed in order to transform the time delays of the network. This transformation allows to change the time delays on the network following some precise rules without affecting the dynamics of the system [7, 8]. The purpose of the transformation is to set $n - 1$ time delays to zero, being n the number of vertices of the network. A brief summary is described in the next section.

Here we take on this idea and propose a new formulation of this transformation that allows to use common optimization algorithms to reduce the number of time delays on a network. Our results show that on networks with different topologies, the number of time delays that can be reduced to zero is larger than $n - 1$. We claim that in most cases the number n_z of zero time delays can be larger than the lower bound $n_z = n - 1$. Moreover, within our framework we can devise other optimization strategies to find a suitable configuration of time delays given a specific need.

The technique described in [8] hinges on the observation that we can change the time delays in a network with a single cycle of length n without altering the dynamics as long as the sum of the time delays around the cycle is conserved. This property can be generalized over arbitrary networks. First, we reformulate the fundamental property of conservation of the time delays over a loop

using algebraic graph theory. The problem of finding minimal time delays on the network is next transformed into a linear optimization problem. We show that the simplex optimization algorithm [9] provides a solution for the transformed time delays where at least $n - 1$ time delays are set to zero.

II. COMPONENTWISE TIME-SHIFT TRANSFORMATION

We consider a graph G with a collection of l oriented edges e_i and n vertices v_i . At each vertex we have a very general dynamical system in the form of a system of n coupled delay differential equations

$$\frac{dx_i}{dt} = f_i(x_i, x_j(t - \tau_k)_{k \in S_i}), \quad (1)$$

with $i = 1, \dots, n$ and S_i is the set of indices k such that the edges e_k connects the vertex j to the vertex i . We assume a discrete time delay τ_k on the edge e_k .

The previous system in Eq. (1) can be transformed with a redefinition of the time delays τ_k without changing the dynamical properties of the system. We set

$$\frac{dy_i}{dt} = f_i(y_i, y_j(t - \tilde{\tau}_k)_{k \in S_i}), \quad (2)$$

with the following change of variables

$$y_i(t) = x_i(t - \eta_i) \quad (3)$$

$$\tilde{\tau}_k = \tau_k + \eta_{s(k)} - \eta_{t(k)}, \quad (4)$$

being η_i constants and $s(k)$ is the source vertex of the edge k and $t(k)$ the target vertex of the same edge. The authors in [8] noticed that the algebraic sum of the time delay around any cycle of the network is constant for every choice of the time-shifts η_i . The term algebraic sum means here that the time delay should be summed up or subtracted according to the relative orientation of the edges around a cycle.

Now the problem is to find the time-shifts η_i associated to each vertex for a desired configuration of time delays $\tilde{\tau}_k$.

III. GRAPH CHARACTERISTICS

The topology of the graph can be described in terms of algebraic structures associated to the topology [10]. We first give some definitions to set the context of the work. We define $G(V, E, A)$ as a directed and connected graph, where V is a set of n vertices and E a set of l directed edges. Before going into the details, we need to renumber the edges from 1 to l and we note τ_k as the time delay of the edge e_k .

To represent the connectivity, we define the incidence matrix $A \in \mathbb{Z}^{n \times l}$ that relates the vertices to the edges. The elements a_{jk} of the matrix A are expressed in the following way: $a_{jk} = 1$ if the edge e_k points towards the vertex j and $a_{jk} = -1$ if the edge points outwards. All other entries are zero. All the information about the connections of the network is contained in this matrix. It is also possible to develop the method for multiple edges connecting two vertices. We restrain here the case to a maximum of two edges to represent a bidirectional connection.

If the graph is connected, or weakly connected, we can define an acyclic subgraph called spanning tree that connects all the vertices and have exactly $n - 1$ edges. This structure is important for the decomposition of the graph G into elementary cycles. Given a spanning tree T and an edge e not in T , there is a unique cycle in G containing only edges of T and e . As a consequence, we can decompose the network into $c = l - (n - 1)$ independent cycles. This decomposition can be expressed as a matrix $B \in \mathbb{Z}^{(l-(n-1)) \times l}$ that expresses the cycle space associated to the tree T . First, we set the orientation of the cycle as the direction of the edge not in T . Being b_{jk} an element of B , we set $b_{jk} = 1$ if the edge k is in the cycle j with the same direction, and $b_{jk} = -1$ if the orientations are opposite. All other numbers are zero. This matrix B is of special interest for our study since the sum of time delays around each cycle is given by a simple matrix multiplication

$$B\tau = \sigma, \quad (5)$$

where $\tau = (\tau_1 \dots \tau_l)^T$ is the column vector of the time delay k associated to the edge e_k and v^T denotes the transpose of the vector v . The vector σ is what matters for the dynamics of the coupled system of delay differential equations. The time delays can be shuffled and changed into a new vector $\tilde{\tau}$, but the vector σ should be constant, so that

$$B\tau = B\tilde{\tau}. \quad (6)$$

This is the key property of the graph that we need to explore the space of possible solutions of $\tilde{\tau}$.

The last necessary step to obtain the full characterization of the system is to derive the time-shifts η_i in Eq. (3) that can lead us back to the time series of the original configuration of Eqs. (1). These time-shifts η_j associated to the vertex j in the network is computed

from a recursive relation on the spanning tree T [8],

$$\tilde{\tau}_k - \tau_k = \eta_{s(k)} - \eta_{t(k)}, \quad (7)$$

with $s(k)$ the source vertex of the edge k and $t(k)$ the target vertex of the same edge. Since the time-shifts are defined up to a constant, we can choose the value $\eta_1 = 0$ as a reference for all the other vertices. The rest of the time-shifts can be obtained from the incidence matrix A restricted to the edges of the spanning tree. The columns of the incidence matrix contain exactly the values $s(k)$ and $t(k)$ for any edge k . If we partition the edges into two subsets of edges in and out of the spanning tree, we can rearrange the incidence matrix in two blocks:

$$A = [A_{in} \mid A_{out}]. \quad (8)$$

The first matrix $A_{in} \in \mathbb{Z}^{n \times (n-1)}$ contains the information on the edges in the spanning tree. We also we split the time delays in two similar sets τ_{in} and τ_{out} and we define a column vector with the time-shifts $\eta = (\eta_1 \dots \eta_n)^T$. From the recursive relation given in Eq. (7), we can infer that

$$\tilde{\tau}_{in} - \tau_{in} = A_{in}\eta. \quad (9)$$

However, we are looking for the time-shifts η as a function of the time delays of the tree T . Noticing that the matrix A_{in} as a rank $n - 1$ and that the vector η has $n - 1$ unknown since $\eta_1 = 0$, we can construct a full rank square matrix A_r by removing the first column of A_{in} . We define the vector $\eta^- = (\eta_2 \dots \eta_n)^T$ and transform the last equation into:

$$\tilde{\tau}_{in} - \tau_{in} = A_r\eta^-. \quad (10)$$

Now we have a fully determined linear system:

$$\eta^- = A_r^{-1}(\tilde{\tau}_{in} - \tau_{in}). \quad (11)$$

The matrices A and B in Eq. (8) and (6) are straightforward to derive and have a strong dependence to each other [11]. Notice also that the matrices A_{in} , A_{out} and B depend on the initial chosen spanning tree. We can demonstrate that this choice does not affect the space of possible solutions that can be reached with Eq. (6). Any spanning tree can give us a valid basis to reconfigure the time delays in the network.

IV. OPTIMIZATION OF NETWORK TIME DELAYS

The main problem is stated in Eq. (6) where all the possible vectors $\tilde{\tau}$ are contained. We have to restrain however the problem to positive time delays τ_k to avoid complications with negative time delays. It consists of finding a vector $\tilde{\tau}$ that will minimize the sum of the time

delays over the network. This problem takes naturally the form of a standard linear program, that is,

$$\begin{aligned} & \text{Minimize: } \sum \tilde{\tau}_k \\ & \text{Constrained to: } B\tilde{\tau} = \sigma \\ & \quad \tilde{\tau}_k \geq 0 \end{aligned} \quad (12)$$

This standard linear program can be solved with conventional techniques such as the simplex optimization algorithm [9]. We show that the simplex method reduces the time delays of the network with at least $n - 1$ zero time delays.

Proposition 1. *Using the simplex algorithm, we can guarantee that there is a feasible solution $\tilde{\tau}$ to the problem in Eq. (12) such that at least $n - 1$ time delays in the vector $\tilde{\tau}$ are set to zero.*

Proof. In the simplex algorithm, there is a first search for a basic feasible solution to the problem in a l -dimensional space. For such a solution, the columns of the matrix B are rearranged into $[D|Z]$ where D is an invertible $c \times c$ matrix and Z is a $c \times (n - 1)$ matrix. The vector $\tilde{\tau} = (\tau_D \tau_Z)$ solution to the equation $B\tilde{\tau} = \sigma$ can be decomposed into $\tau_D = D^{-1}\sigma$ and $\tau_Z = \mathbf{0}$ a vector with all zeros. Being $n - 1$ the size of the vector τ_Z , we have a valid reduction of the network with $n - 1$ time delays set to zero. The other part τ_D contains only positive time delays. \square

The existence of one basic feasible solution gives us a valid reduction, however the algorithm looks further for an optimal solution minimizing the sum of the time delays. The solver will find in general a better solution with more time delays equal to zero or at least with a total sum of the time delays below or equal than the initial sum of the time delays $\sum \tau_k$. There are plenty of efficient implementations of the simplex algorithm to solve linear programs [9], and we can obtain the reduction of the network in a polynomial time.

We now have all the ingredients to construct a optimized network. All we need is any spanning tree T , the incidence matrix A and a fundamental loop matrix B of the graph G .

V. NUMERICAL EXPERIMENTS

Here we apply the previous optimization method to networks with different topologies and we measure the efficiency of the optimal solution of the time delay reduction. We focus on three popular random networks: Erdős-Renyi, small-world and scale-free networks [12, 13]. We distribute the time delays randomly across the network using two types of statistical distributions: discrete and continuous uniform distributions. The discrete distribution consists in picking randomly integers from 1 to 10. The continuous distribution is a uniform random distribution in $[0, 10]$. We apply the algorithm

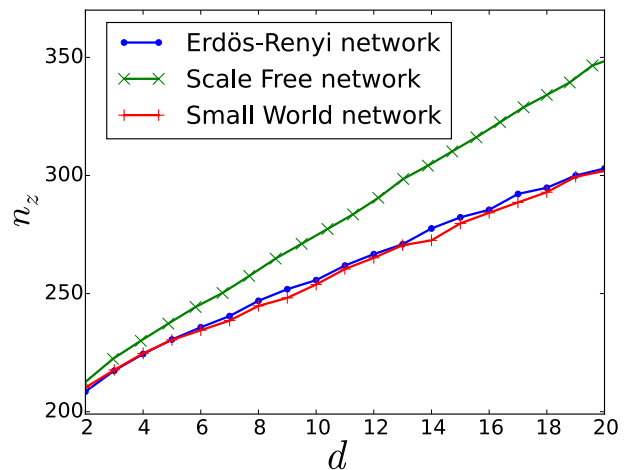


FIG. 1. **Reduction of time delays for different network topologies.** Number of zero time delays n_z against the mean degree of the vertices in the network d . The scale-free network presents a clear improvement in terms of n_z while there is almost no difference between the Erdős-Renyi and the small-world topology

to networks with $n = 200$ vertices and a number of edges that varies with the mean vertex degree d . The results have been averaged over 100 network realizations. We first detail the results for the discrete time delay distribution on the three types of topology.

The numerical optimization shows some unexpected results. First, the optimal solution is different depending on the topology of the network. The scale-free topology gives better results in the sense that n_z is higher for this topology for a given value of d . On the other hand, the random Erdős-Renyi and small-world network give similar results. In the three cases the solution is on average well above the lower bound $n_z = n - 1 = 199$. The source of the differences between the distinct cases is unclear, however it is seemingly linked to the topology of the networks.

However, this improvement in n_z drops dramatically if we consider a continuous distribution or a discrete distribution with a large number of elements. When the time delays are distributed following a continuous distribution, it is almost impossible to find a solution with $n_z > n - 1$. The simplex method finds only the basic feasible solution $n_z = n - 1$. The reason is that it is difficult to find a linear combination of random real numbers equal to zero. On the contrary, it happens much more frequently when the time delays are taken from a small set of integers.

If we measure the ratio r between the total sum of the time delays before and after the optimization defined as

$$r = \frac{\sum_{k=1}^l \tilde{\tau}_k}{\sum_{k=1}^l \tau_k}, \quad (13)$$

we can see that there is a substantial reduction in the process. If we look closer at the simulations shown in

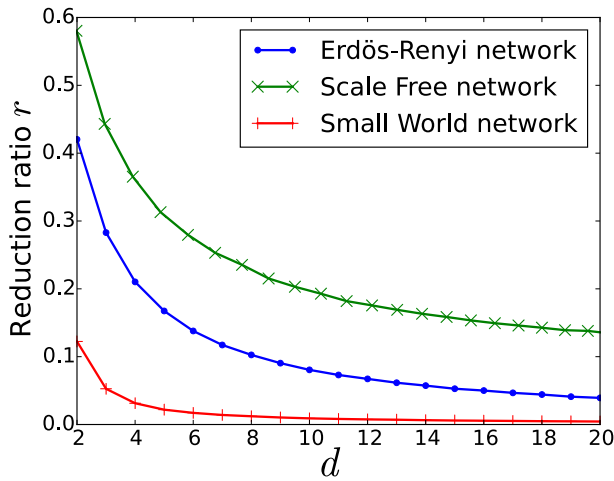


FIG. 2. **Reduction ratio for the three topologies.** Reduction ratio computed as $r = \sum \tilde{\tau}_k / \sum \tau_k$. There are significant differences between the distinct topologies, being the small-world topology the worse candidate.

Fig. 2, the improvement also varies from one distribution to another. As the average degree d increases, it also seems that it is difficult to improve the ratio r .

The previous examples focus on the properties of the networks and do not involve any specific dynamical system. We present an application where a network of Kuramoto phase oscillators is coupled with time delays [4]. The phase oscillator model is a very simple abstraction of the essential properties of limit cycle oscillators. We can use this model to test our optimization method on a complex network of simple dynamical systems. The setup consists of a unidirectional Erdős-Renyi network with average degree d , where the vertices represent Kuramoto oscillators with an identical intrinsic frequency ω . The edges of the network represent a time delayed interaction chosen randomly according to a statistical distribution. The coupled delay differential equation can be written as

$$\frac{d\theta_i}{dt} = \omega + \frac{K}{d} \sum_{k \in S_i} (\theta_j(t - \tau_k) - \theta_i), \quad (14)$$

where S_i is the set of edges inciding from vertex j to the vertex i and K is the coupling strength. We distribute the time delays τ_k following a uniform distribution in the continous interval $[\tau_m, 0.5 + \tau_m]$. Notice however that since we integrate the equation numerically, we have to discretize this interval due to the finite time step size of the algorithm. In order to test the validity of the reduction in a dynamical system, we use the average frequency of the network since this measurement is independent of the initial history of the delay differential equation [14].

We let evolve the network in time and we compute the average frequency Ω_i of each oscillator over a finite

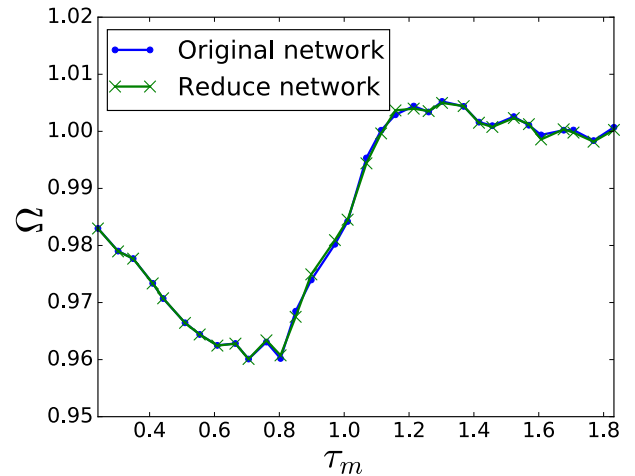


FIG. 3. **Average network network frequency Ω of a coupled network of Kuramoto phase oscillators coupled with time delays.** The curves, that are superposed, represent the average network frequency for the original (dot markers) and reduced network (cross markers). For each dot the average network frequency has been computed and averaged for several initial histories of the network to avoid numerical artifacts caused by the integration method. Both the original and reduced network lead to the same asymptotic frequency. Parameters are: $\omega = 1$, $K = 0.1$, $\tau_k \in [\tau_m; \tau_m + 0.5]$, $n = 50$, $d = 4$.

interval of time T

$$\Omega_i = \frac{1}{T} \int_0^T \dot{\theta}_i dt. \quad (15)$$

Then we compute the average network frequency Ω in this manner

$$\Omega = \frac{1}{n} \sum_i \Omega_i. \quad (16)$$

This last frequency is independent of the chosen initial conditions and should be the same for both the original network and the reduced network given by Eq. (12). In Fig. 3, we show an example where a network of $n = 50$ oscillators has been simulated with a realization of the random time delays. The average frequency of the original and reduced network are consistent in both simulations showing that the asymptotic behavior is the same. Other measures such as the order parameter are unsuitable to make comparisons since the time-shifts η_i will alter the value.

The simulations have been performed with the programming language Julia [15] using LightGraphs, JuMP and Coin-or Linear Programming (Clp) packages.

VI. CONCLUSIONS

Reorganizing the time delays in a network does not seem an easy task at first sight. But once the basic

mechanisms of time delay conservation are understood, it is possible to change the time delays and at the same time conserving the dynamical properties of the network. Our formulation along with the componentwise time-shift transformation technique opens a way to reduce even further the time delay space. When the problem is stated in the form of a linear program, the simplex algorithm gives in general a higher number of zero time delays than the theoretical lower bound n_z , that corresponds to the dimension of the cycle space of the network. It also finds the solution with the lowest sum of time delays, which can represent a reduction up to 60% of the initial sum of the time delays.

This work points into a new direction for speeding up the numerical integration of coupled dynamical systems with time delays. The presence of different time delays among the network usually involves a high computational

and storage cost. It may be possible to optimize the network to reduce this computational load. Another possible application is to modify the fitness function of the optimization algorithm such that the time delays fit a desired distribution more suitable to the problem at glance.

It has been shown in [16] that the cyclic motifs play an important role in the dynamics of networks of coupled phase oscillators with time delay. It is an interesting observation when contrasted with the fact that in this paper the time delay around fundamental loops plays a central role in the dynamics.

ACKNOWLEDGEMENTS

Financial support from the Spanish Ministry of Economy and Competitiveness under Project No. FIS2016-76883-P is acknowledged.

-
- [1] Soriano, M. C., García-Ojalvo, J., Mirasso, C. R. & Fischer, I. Complex photonics: Dynamics and applications of delay-coupled semiconductor lasers. *Rev. Mod. Phys.* **85**, 421 (2013).
 - [2] Liang, X., Tang, M., Dhamala, M. & Liu, Z. Phase synchronization of inhibitory bursting neurons induced by distributed time delays in chemical coupling. *Phys. Rev. E* **80**, 066202 (2009).
 - [3] Perez, T. *et al.* Effect of the topology and delayed interactions in neuronal networks synchronization. *PLoS ONE* **6**, e19900 (2011).
 - [4] Yeung, M. S. & Strogatz, S. H. Time delay in the kuramoto model of coupled oscillators. *Physical Review Letters* **82**, 648 (1999).
 - [5] Earl, M. G. & Strogatz, S. H. Synchronization in oscillator networks with delayed coupling: A stability criterion. *Physical Review E* **67**, 036204 (2003).
 - [6] Lee, W. S., Ott, E. & Antonsen, T. M. Large coupled oscillator systems with heterogeneous interaction delays. *Phys. Rev. Lett.* **103**, 044101 (2009).
 - [7] Lücken, L., Pade, J. P., Knauer, K. & Yanchuk, S. Reduction of interaction delays in networks. *EPL* **103**, 10006 (2013).
 - [8] Lücken, L., Pade, J. & Knauer, K. Classification of Coupled Dynamical Systems with Multiple Delays: Finding the Minimal Number of Delays. *SIAM Journal on Applied Dynamical Systems* **14**, 286–304 (2015).
 - [9] Bazaraa, M. S., Jarvis, J. J. & Sherali, H. D. *Linear programming and network flows* (John Wiley & Sons, 2011).
 - [10] Biggs, N. *Algebraic Graph Theory* (Cambridge university press, 1993).
 - [11] Deo, N. *Graph Theory with Applications to Engineering and Computer Science* (Courier Dover Publications, 2016).
 - [12] Small, M., Hou, L. & Zhang, L. Random complex networks. *National Science Review* **1**, 357 (2014).
 - [13] Newman, M. *Networks: an introduction* (Oxford university press, 2010).
 - [14] Nordenfelt, A., Wagemakers, A. & Sanjuán, M. A. F. Frequency dispersion in the time-delayed kuramoto model. *Physical Review E* **89**, 032905 (2014).
 - [15] Bezanson, J., Karpinski, S., Shah, V. B. & Edelman, A. Julia: A fast dynamic language for technical computing. *arXiv*, 1209.5145 (2012).
 - [16] Nordenfelt, A., Wagemakers, A. & Sanjuán, M. A. F. Cyclic motifs as the governing topological factor in time-delayed oscillator networks. *Physical Review E* **90**, 052920 (2014).